

**VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY
UNIVERSITY OF INFORMATION TECHNOLOGY**

---oOo---



**Thị giác máy tính
CS231.P11**

**ĐỒ ÁN MÔN HỌC: PHƯƠNG PHÁP XÁC ĐỊNH
VẬT THỂ TRÊN ẢNH (Object Detection)**

STUDENT NAME:

Nguyễn Quang Dũng – 22520286

Trần Trí Dũng – 22520293

Trần Nguyễn Thanh Duy – 22520344

Lecturer: Mai Tiến Dũng

Mục lục

1. Phát biểu bài toán, input và output	3
2. Lý do	5
3. Phương pháp	5
3.1. Bài toán Classification detection	6
3.1.1. Đặc trưng	6
3.1.1.1. Đặc trưng canny egde:	6
3.1.1.2. Đặc trưng HOG	7
3.1.2. Thuật toán phân loại:	8
3.1.2.1. KNN (K-Nearest Neighbors)	8
3.1.2.2. Logistic Regression	10
3.1.2.3. SVM (Support Vector Machine)	10
3.2. Bài toán Localization	12
3.2.1. Sliding window	12
3.2.2. Non – Maximum Suppression	13
4. Kết luận	14
5. Thông tin và tiến độ thành viên	15
Tài liệu tham khảo:	15
Source code:	15
Một số dự đoán:	16

1. Phát biểu bài toán, input và output

Ở đồ án này, nhóm xây dựng phương pháp để xác định vật thể trong bức ảnh, vật thể ở đây có thể khác nhau tùy thuộc vào tập dữ liệu huấn luyện được đưa vào.

Về bản chất thì có thể chia bài toán này thành hai bài toán nhỏ đặc trưng trong thị giác máy tính:

- Bài toán phát hiện vật thể (Classification)
- Bài toán định vị vật thể (Localization)

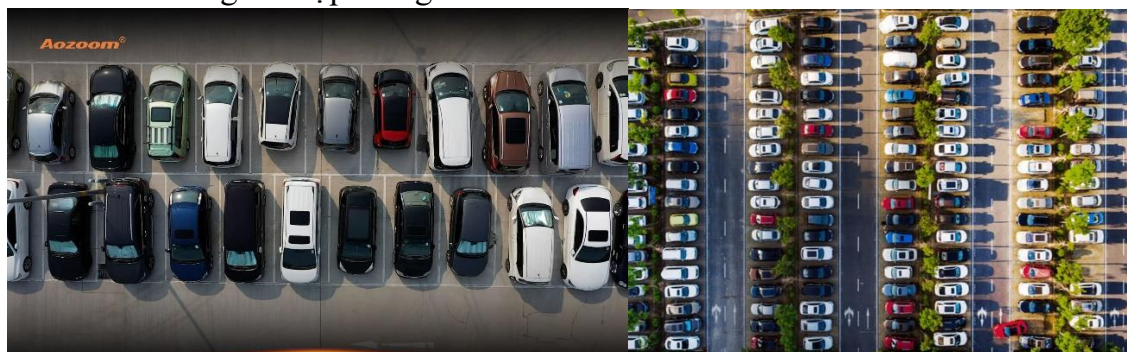
Trong khuôn khổ đồ án này, nhóm chỉ sử dụng những thuật toán phân loại đơn giản (KNN, SVM, Logistic Regression) cùng với phương pháp trích xuất đặc trưng phù hợp cho bài toán (Classification), sử dụng phương pháp sliding window cho bài toán Localization.

Input và output cho bài toán tổng quát lớn có thể được phát biểu như sau:

- Input:

Bức ảnh có chứa vật thể (nhiều hoặc ít) bên trong, góc chụp cố định (sự thay đổi hình dạng của vật thể theo thời gian không đáng kể)

Vd1: bãi đỗ xe góc chụp thẳng từ trên cao



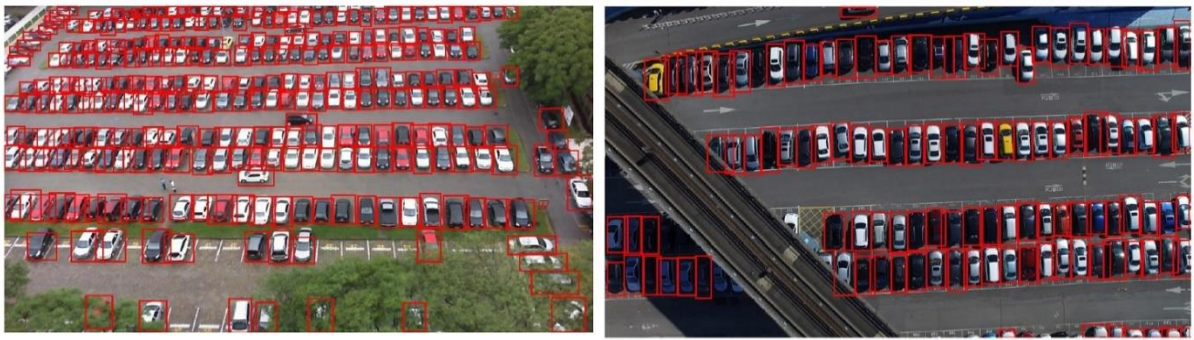
Vd2: ảnh chụp đầu của các ống nước



- Output:

Vị trí của vật thể phát hiện được trong bức ảnh

Vd:



Đối với bài toán localization, input và output có thể được phát biểu như sau:

- Input:

Ảnh $x \Rightarrow \text{List}x = \{x_i\}_N$ thuộc x

- Output:

(bounding box) $\text{List}y = \{y_i\} = f_{\text{local}}(f_{\text{Classification}}(\{x_i\})) = f_{\text{local}}(\text{List}y_{\text{Classification}})$

Đối với bài toán Classification, input và output có thể được phát biểu như sau:

- Input:

$D = \{(x_i, y_i)\}$

$L = U_{y_i} = \{\text{có vật thể, không có vật thể}\}$

ảnh x (vector đặc trưng)

Vd:



- Output:

(predicted class, probability) $y_{\text{Classification}} = f_{\text{Classification}}(x)$ thuộc L

Về cơ bản, từ một bức hình lớn ban đầu, thông qua bài toán Local sẽ có được danh sách những bức hình nhỏ hơn được cắt từ bức hình lớn.

Các bức hình nhỏ hơn sẽ thông qua bài toán Classification và có được kết quả dự đoán cho bức hình nhỏ.

Từ danh sách kết quả dự đoán của các bức hình nhỏ, bài toán Local sẽ lọc và trả về sau cùng những bức hình nhỏ đã đủ điều kiện (tọa độ Bounding box), chính là output sau cùng.

2. Lý do

Nhu cầu thực tế:

a. Quản lý và giám sát:

- Đếm xe giúp quản lý giao thông, giảm ùn tắc, tối ưu đèn tín hiệu.
- Đếm sản phẩm hỗ trợ giám sát năng suất và phát hiện lỗi sản xuất.

b. An ninh và phân tích hành vi:

- Đếm người giúp phát hiện đám đông bất thường, đảm bảo an ninh.

Hỗ trợ ra quyết định:

a. Tăng hiệu quả:

- Đếm vật thể giúp tối ưu nguồn lực, cảnh báo tự động.
- Ví dụ: Điều chỉnh nhân viên bán lẻ dựa trên số lượng khách hàng

b. Dự đoán:

- Cung cấp dữ liệu phân tích xu hướng, dự báo nhu cầu.

Ứng dụng khoa học:

a. Robot học:

- Robot đếm hàng trong kho đảm bảo độ chính xác khi vận chuyển.
- b. Phân tích môi trường:
- Đếm động vật hoặc cây cối từ ảnh vệ tinh đánh giá tác động môi trường.

Phát triển AI:

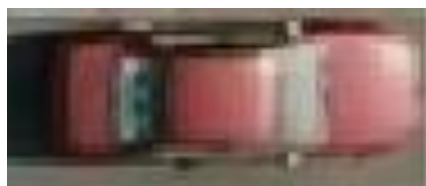
- Trên thực tế, hiệu năng sử dụng phương pháp này để đếm vật thể với số lượng lớn vẫn chưa được tối ưu, vì tốc độ xử lý vẫn còn tương đối chậm trong khi độ chính xác và mức độ hiệu quả của đặc trưng vẫn thua xa so với các kỹ thuật học sâu, tuy nhiên, nhóm nghĩ đây vẫn một phương pháp tiềm năng đối với những bài toán cần nhận diện và định vị vật thể đơn giản hơn.

3. Phương pháp

Vì mục tiêu của nhóm là nhận diện nhiều loại vật thể khác nhau một cách riêng lẻ, tùy vào tập dataset đưa vào, vì thế đặc trưng nhóm hướng đến là đặc trưng về hình dạng, cạnh, không phụ thuộc vào thông tin màu.

3.1. Bài toán Classification detection

3.1.1. Đặc trưng



3.1.1.1. Đặc trưng canny egde:

Canny Edge Detection là một phương pháp phát hiện biên trong ảnh, được giới thiệu bởi John F. Canny vào năm 1986. Đây là một kỹ thuật dựa trên toán học và tối ưu hóa, được thiết kế để phát hiện các cạnh rõ ràng và giảm nhiễu trong ảnh.

Cách hoạt động:

1. Làm mờ ảnh: Ảnh được làm mờ bằng bộ lọc Gaussian để giảm nhiễu.
2. Tính gradient: Xác định độ lớn và hướng gradient tại mỗi điểm ảnh bằng cách sử dụng các đạo hàm.
3. Non-maximum suppression: Loại bỏ các điểm không phải là cực đại cục bộ để làm mỏng các cạnh.
4. Double thresholding: Phân loại cạnh mạnh (giữ lại), cạnh yếu (xét tiếp), và loại bỏ nhiễu dựa trên hai ngưỡng giá trị.
5. Edge tracking by hysteresis: Liên kết các cạnh yếu với các cạnh mạnh nếu chúng liền kề, tạo thành các cạnh đầy đủ.

Canny edge nổi bật nhờ khả năng cân bằng giữa độ chính xác và khả năng chống nhiễu tốt.

Trong **thực nghiệm** của nhóm, đặc trưng Canny edge được cải thiện bằng cách:

- Mở rộng biên để làm rõ các cạnh.
- Loại bỏ các khe hở nhỏ trong đường biên



Accuracy KNN, độ đo tương quan:

	Chỉ canny edge	CE đã chỉnh sửa
Car k = 10	0.84	0.83
Pipe k = 8	0.81	0.88
Human k = 9	0.68	0.64

3.1.1.2. Đặc trưng HOG

HOG (Histogram of Oriented Gradients) là một đặc trưng thường dùng trong xử lý ảnh và thị giác máy tính để phát hiện đối tượng, được giới thiệu bởi Navneet Dalal và Bill Triggs vào năm 2005 trong bài báo về phát hiện người.

Hoạt động của HOG:

1. Phân vùng ảnh: Ảnh được chia thành các ô nhỏ (cells).
2. Tính gradient: Tính toán độ lớn và hướng của gradient tại mỗi pixel để xác định cách thay đổi cường độ sáng.
3. Lập histogram: Với mỗi ô, histogram được tạo dựa trên các hướng gradient.
4. Chuẩn hóa: Để tăng tính ổn định, các histogram trong khối (block) lân cận được chuẩn hóa, làm giảm ảnh hưởng của thay đổi ánh sáng.

Ưu điểm:

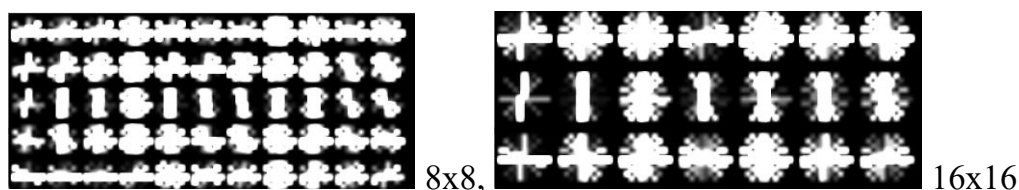
- Nổi bật trong việc phát hiện các mẫu hoặc hình dạng vật thể

pixels_per_cell (số pixel trên mỗi ô):

Đây là kích thước của mỗi ô (cell) trong đơn vị pixel. Ví dụ, nếu `pixels_per_cell=(8, 8)`, điều đó có nghĩa là mỗi ô sẽ gồm 8 pixel theo chiều ngang và 8 pixel theo chiều dọc. Trong mỗi ô, một histogram sẽ được tính toán dựa trên hướng gradient của các pixel.

cells_per_block (số ô trên mỗi khối):

Đây là kích thước của mỗi khối (block) trong đơn vị số lượng ô. Mỗi khối chứa nhiều ô và được sử dụng để chuẩn hóa giá trị của histogram trong các ô nhằm giảm nhiễu và tăng độ chính xác. Ví dụ, nếu `cells_per_block=(2, 2)`, điều đó có nghĩa là mỗi khối sẽ bao gồm 2 ô theo chiều ngang và 2 ô theo chiều dọc.



So sánh accuracy (KNN, độ đo tương quan):

Đặc trưng	Canny edge	HOG 8x8	HOG 16x16
Car	0.83 (k = 10)	0.95 (k = 4)	0.93 (k = 4)
Pipe	0.88 (k = 8)	0.92 (k = 4)	0.94 (k = 4)
Human	0.64 (k = 9)	0.88 (k = 4)	0.89 (k = 4)

3.1.2. Thuật toán phân loại:

3.1.2.1. KNN (*K-Nearest Neighbors*)

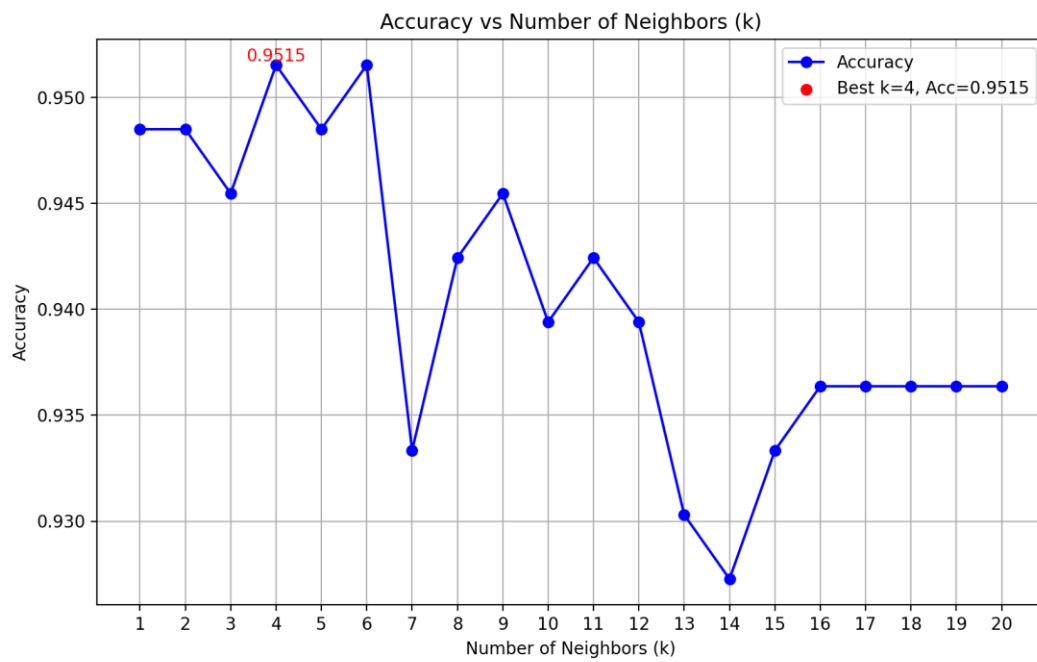
- **Nguyên lý:** KNN là thuật toán phân loại hoặc hồi quy dựa trên khoảng cách. Một điểm mới được gán nhãn hoặc giá trị dựa vào K điểm lân cận gần nhất trong tập huấn luyện.
- Cách hoạt động:
 1. Tính khoảng cách giữa điểm mới và tất cả các điểm trong tập dữ liệu
 2. Chọn K điểm gần nhất.
 3. Dựa vào đa số nhãn (phân loại) hoặc giá trị trung bình (hồi quy) của các điểm này để dự đoán.
- **Ưu điểm:** Đơn giản, không yêu cầu huấn luyện.
- **Nhược điểm:** Hiệu quả phụ thuộc vào giá trị K và cách đo khoảng cách; tốn tài nguyên nếu dữ liệu lớn.
- **Thực nghiệm:** đối với thuật toán KNN này, nhóm thử nghiệm nhiều tham số K và độ đo, trong số đó, độ đo tương quan (correlation) cho kết quả tốt hơn độ đo mặc định thường thấy là Euclid.

Độ đo

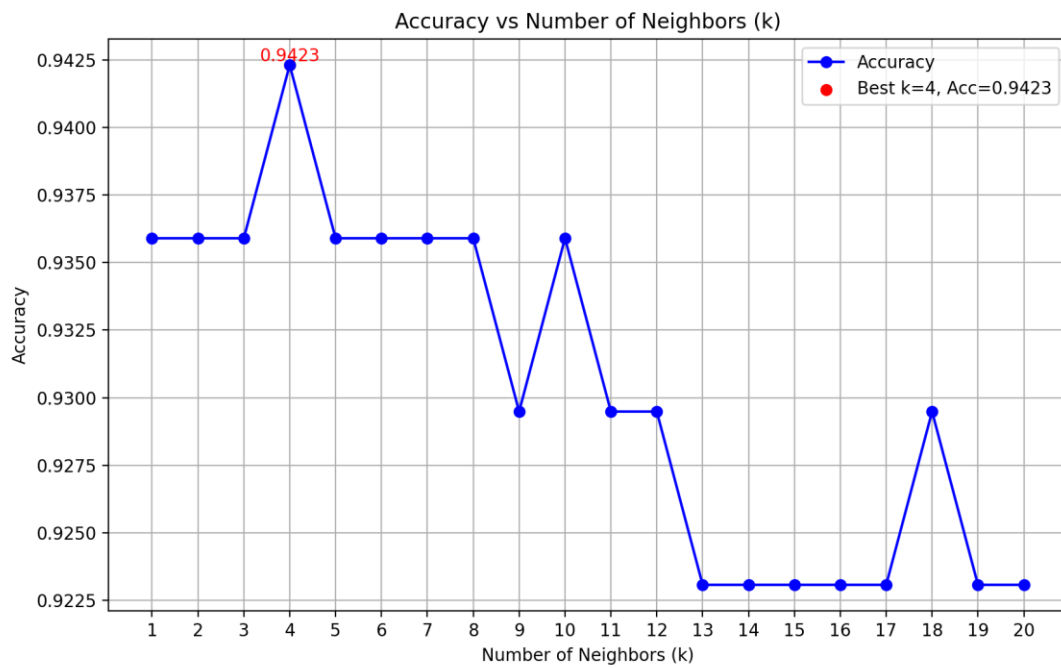
	Euclid	Correlation
Car, HOG8x8, KNN k=4	0.9	0.95
Pipe, HOG16x16, KNN k = 4	0.93	0.94
Human, HOG16x16, KNN k = 4	0.79	0.89

Sử dụng độ đo tương quan cho thuật toán KNN.

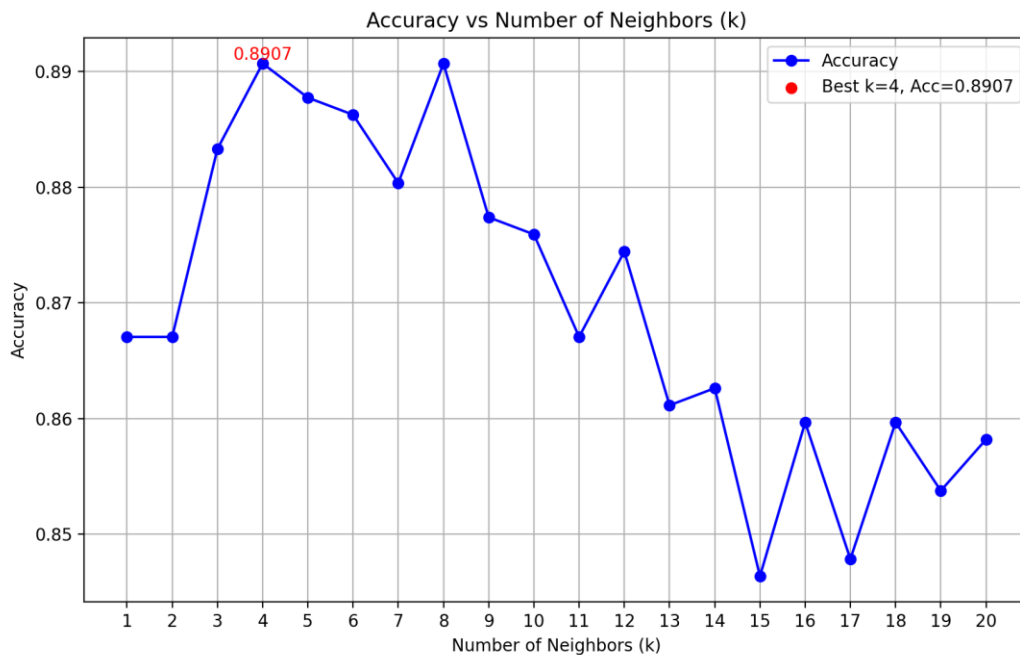
Car, HOG8x8:



Pipe, HOG16x16:



Human, HOG16x16:



3.1.2.2. Logistic Regression

- **Nguyên lý:** Logistic Regression là một thuật toán phân loại dựa trên hồi quy tuyến tính nhưng sử dụng hàm sigmoid để chuyển đổi giá trị dự đoán thành xác suất.
- **Cách hoạt động:**
 1. Xác định mối quan hệ giữa đầu vào X và đầu ra y qua hàm tuyến tính.
 2. Dùng hàm sigmoid σ để tính xác suất.
 3. Gán nhãn dựa trên ngưỡng.
- **Ưu điểm:** Dễ hiểu, hiệu quả cho phân loại nhị phân và có thể mở rộng sang phân loại đa nhãn (One-vs-Rest).
- **Nhược điểm:** Giới hạn với bài toán tuyến tính, không tốt khi dữ liệu không tách rời được.
- **Thực nghiệm:** Đối với phương pháp này, nhóm sử dụng tham số mặc định được cài đặt của Sklearn.

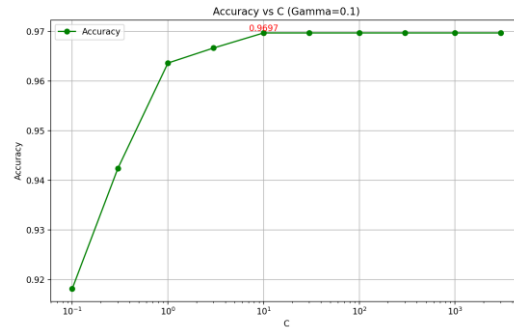
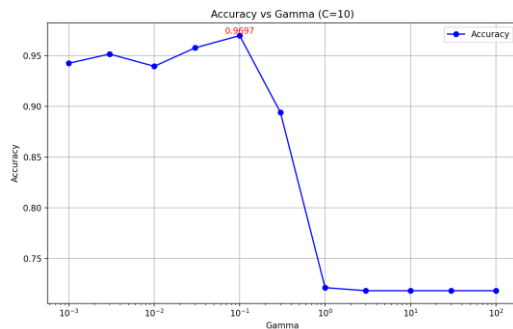
3.1.2.3. SVM (Support Vector Machine)

- **Nguyên lý:** SVM phân loại dữ liệu bằng cách tìm siêu phẳng (hyperplane) tối ưu, tối đa hóa biên (margin) giữa các lớp.
- **Cách hoạt động:**
 1. Chọn một siêu phẳng tách rời các điểm thuộc hai lớp (tuyến tính).

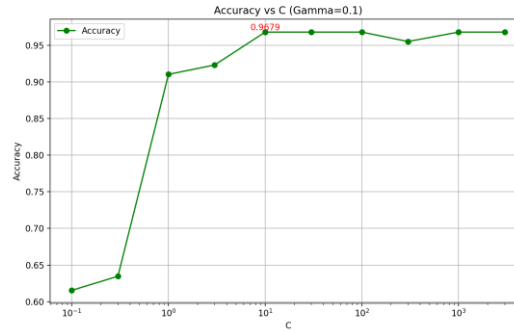
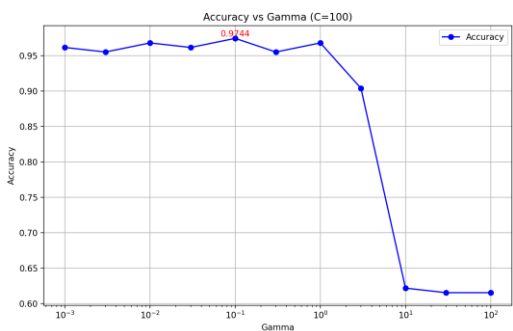
2. Với dữ liệu không tuyến tính, sử dụng kernel trick để ánh xạ dữ liệu sang không gian cao hơn, giúp tách rời dễ hơn.

- **Ưu điểm:** Hiệu quả cao với dữ liệu nhỏ và tách biệt rõ ràng; linh hoạt với kernel.
- **Nhược điểm:** Không phù hợp với dữ liệu lớn.
- **Thực nghiệm:**

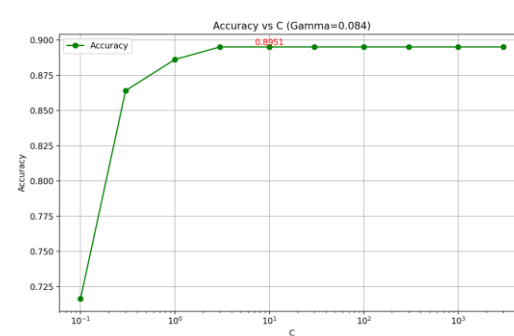
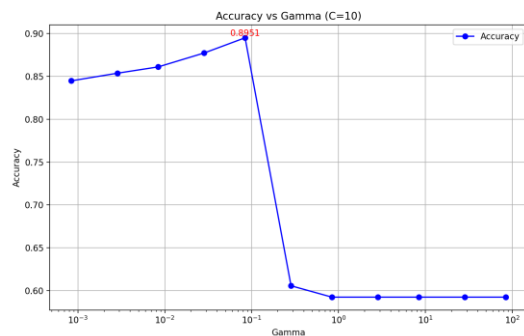
Car:



Pipe:



Human



So sánh:

Thuật toán	KNN	LR	SVM
Car	Accuracy: 0.9515151515151515	Accuracy: 0.9515151515151515	Accuracy: 0.9696969696969697

	Precision: 0.9377379619260918 Recall: 0.9515151515151515 Confusion matrix: [[86 7] [9 228]]	Precision: 0.9401116101810263 Recall: 0.9515151515151515 Confusion matrix: [[85 8] [8 229]]	Precision: 0.9686232458389669 Recall: 0.9696969696969697 Confusion matrix: [[86 7] [3 234]]
Pipe	Accuracy: 0.9423076923076923 Precision: 0.9351648351648352 Recall: 0.9423076923076923 Confusion matrix: [[58 2] [7 89]]	Accuracy: 0.9615384615384616 Precision: 0.9571036376115305 Recall: 0.9615384615384616 Confusion matrix: [[58 2] [4 92]]	Accuracy: 0.9743589743589743 Precision: 0.9704873026767331 Recall: 0.9743589743589743 Confusion matrix: [[59 1] [3 93]]
Human	Accuracy: 0.8906942392909897 Precision: 0.8984035527650696 Recall: 0.8906942392909897 Confusion matrix: [[220 56] [18 383]]	Accuracy: 0.8508124076809453 Precision: 0.848344756239493 Recall: 0.8508124076809453 Confusion matrix: [[217 59] [42 359]]	Accuracy: 0.8951255539143279 Precision: 0.8964780224886163 Recall: 0.8951255539143279 Confusion matrix: [[230 46] [25 376]]

Nhận xét:

Ở cả ba tập dữ liệu, đặc trưng HOG cho ra kết quả tốt hơn so với đặc trưng Canny Edge khi xét trên cùng một phương pháp KNN với tham số K đã được tối ưu cho từng kiểu đặc trưng, độ đo tương quan. Đối với thuật toán phân loại, KNN cho ra mô hình có thời gian thực thi tương đối chậm khi so với hai thuật toán còn lại, đồng thời Logistic Regression và SVM cho mô hình có độ chính xác tương đối cao hơn so với SVM, trong đó SVM có phần chính hơn với những tham số đã được tối ưu.

3.2. Bài toán Localization

3.2.1. Sliding window

quét qua toàn bộ ảnh để phân tích từng vùng nhỏ, nhằm tìm kiếm đối tượng. Vì đối tượng có thể xuất hiện ở nhiều vị trí và kích thước khác nhau trong khung hình, kỹ thuật này chia nhỏ và quét ảnh theo từng bước.

Quy trình:

1. Chia ảnh thành các vùng nhỏ (sub-windows):

- Một cửa sổ (window) với kích thước cố định sẽ được di chuyển từ góc trên bên trái của ảnh đến góc dưới bên phải.
- Cửa sổ này "trượt" qua ảnh theo một bước nhảy cố định (step_size), cả theo chiều ngang và chiều dọc.

2. Lấy từng vùng ảnh:

- Tại mỗi vị trí của cửa sổ, vùng ảnh tương ứng được cắt ra.
- Các vùng này có thể được thay đổi kích thước để phù hợp với đầu vào của mô hình.

3. Lặp lại với nhiều kích thước cửa sổ khác nhau:

- Vì đối tượng có thể có kích thước lớn/nhỏ, bạn cần dùng nhiều kích thước cửa sổ (multi-scale sliding window) để quét.

3.2.2. Non – Maximum Suppression


Trong đồ án này, phương pháp non – max suppression được nhóm tự cài đặt để thuận tiện trong việc điều chỉnh.

Mục đích:

Khi sử dụng sliding window, nhiều vùng có thể chồng lấn nhau và cùng phát hiện một đối tượng. **Non-Maximum Suppression** được dùng để loại bỏ các vùng chồng lấn này, chỉ giữ lại vùng có xác suất cao nhất.

Quy trình:

1. Sắp xếp bounding box theo xác suất:
 - Mỗi bounding box có một "điểm số" xác suất (ví dụ: khả năng là xe hơi). Sắp xếp các box theo thứ tự giảm dần của xác suất.
2. Lấy box có xác suất cao nhất:
 - Chọn box đầu tiên trong danh sách (xác suất cao nhất) và thêm nó vào danh sách "giữ lại".
3. Loại bỏ các box chồng lấn:
 - Với các box còn lại, tính Intersection over Union (IoU):
 - Intersection: Phần diện tích giao nhau giữa 2 box.
 - Union: Tổng diện tích của 2 box, trừ đi phần giao.
 - $\text{IoU} = \text{Intersection} / \text{Union}$

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$


- Loại bỏ các box có IoU lớn hơn ngưỡng (threshold). Điều này nghĩa là nếu box nào chồng lấn quá nhiều với box đã chọn, nó sẽ bị loại bỏ.

4. Lặp lại:

- Tiếp tục với box có xác suất cao tiếp theo và lặp lại quá trình cho đến khi không còn box nào.

Thực nghiệm (IoU):

Car (SVM gamma = 0.1, C = 10, HOG8x8)	0.75
Pipe (SVM gamma = 0.1, C = 100, HOG16x16)	0.63
Human (SVM gamma = 0.084, C = 10, HOG16x16)	0.75

Một số detect tính toán IoU:



4. Kết luận

Vì các vật thể khác nhau có kích thước và hình dạng khác nhau, đồng thời kích thước cho bounding box của vật thể đó cũng không tương đồng, vì thế phải tác và thực hiện phân loại các vật thể khác nhau một cách riêng lẻ thay vì làm hết trên một mô hình.

Cần được tối ưu hơn ở khâu train mô hình (bài toán phân loại và bài toán định vị hoạt động vẫn còn tương đối độc lập). Xây dựng lại tập dataset với định dạng phù hợp hơn).

Hiệu năng thấp, không hiệu quả đối với những bài toán đếm những vật thể có hình dạng phức tạp. Để giải quyết, cần tối ưu phép trích xuất đặc trưng đối với mỗi loại vật thể khác nhau (xây dựng phương pháp chuyên biệt cho từng loại vật thể).

Ồn hơn nếu sử dụng trên những vật thể đơn giản như ống nước, đồng xu, ...

5. Thông tin và tiến độ thành viên

Tên – MSSV	Phân công	KQ
Nguyễn Quang Dũng – 22520286	Xây dựng bài toán, làm dataset, làm bài toán local.	10/10
Trần Trí Dũng – 22520293	Làm dataset, làm các phương pháp trích xuất đặc trưng.	10/10
Trần Nguyễn Thanh Duy – 22520344	Làm dataset, làm các thuật toán phân loại.	10/10

Tài liệu tham khảo:

<https://ieeexplore.ieee.org/document/1467360>

<https://ieeexplore.ieee.org/document/4767851>

<https://arxiv.org/abs/1705.02950>

<https://scikit-learn.org/1.5/modules/generated/sklearn.neighbors.KNeighborsClassifier.html>

https://scikit-learn.org/1.5/modules/generated/sklearn.linear_model.LogisticRegression.html

<https://scikit-learn.org/1.5/modules/generated/sklearn.svm.SVC.html>

<https://www.youtube.com/watch?v=F-884J2mnOY>

Source code:

https://github.com/DungQuangUiT/Object_counting

Một số dự đoán:

Car (SVM gamma = 0.1, C = 10, HOG8x8)



Pipe (SVM gamma = 0.1, C = 100, HOG16x16)



Human (SVM gamma = 0.084, C = 10, HOG16x16)

